

# 쓰레디드 메모리 모듈의 메모리 접근 지연 시간 분석

최효진, 성원용

서울대학교 전기공학부

e-mail : [hjchoi@dsp.snu.ac.kr](mailto:hjchoi@dsp.snu.ac.kr), [wysung@snu.ac.kr](mailto:wysung@snu.ac.kr)

## Analyzing Memory Access Latency of Threaded Memory Modules

Hyojin Choi, Wonyong Sung  
School of Electrical Engineering  
Seoul National University

### Abstract

Threaded memory modules can increase the number of memory requests to serve concurrently by simply modifying conventional memory modules such as DIMM (Dual In-line Memory Module). In this paper, the memory access latency of the threaded memory modules is analyzed by using a simulation method. The threaded memory modules cannot enhance the memory access performance in terms of average memory access latency, whereas it is power efficient as reported in other literature [1][2].

### I. 서론

멀티코어 프로세서는 저전력, 연산 능력 향상 등의 장점으로 일반 PC 뿐만 아니라 임베디드 시스템에서도 널리 사용되고 있다. 최근에는 연산 속도를 빠르게 하기 위하여 프로세서 코어의 개수를 증가시키고 있는 추세이다. 그러나, 프로세서 코어의 개수가 증가함에 따라 메인 메모리 접근 지연 시간의 증가로 인해 전체 시스템의 성능 향상이 제한된다. 따라서, 메인 메모리 접근 성능을 향상 시킬 수 있는 방안이 필요하다.

멀티코어 프로세서 시스템의 메모리 접근 성능을 향상시키기 위한 방안 중 하나로 쓰레디드 메모리 모듈(threaded memory module)이 제안된 바 있다[1][2]. 쓰레디드 메모리 모듈은 기존의 DRAM 칩을 그대로 사용하면서 메모리 모듈에 컨트롤 신호를 추가하여 독립적으로 접근할 수 있는 DRAM 칩의 개수를 증가시킨 방식이다. [1]에서는 쓰레디드 메모리 모듈의 데이터 버스 사용량(bus utilization)을 40% 이상 향상 시

킬 수 있음을 보였다. [2]에서는 전력 소모 감소 효과를 시뮬레이션을 통해 보였다. 본 연구에서는 멀티코어 프로세서 시스템에서 쓰레디드 메모리 모듈의 메모리 접근 시간을 시뮬레이션 방법으로 분석하였다.

### II. 본론

#### 2.1 쓰레디드 메모리 모듈

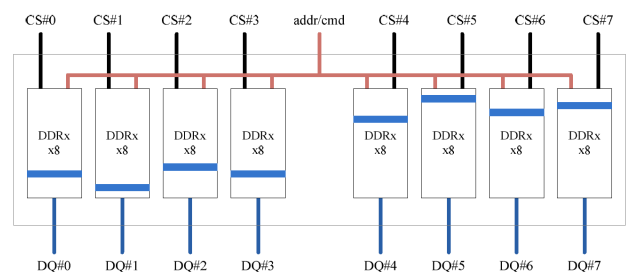


그림 1. 쓰레디드 메모리 모듈

기존의 메모리 모듈은 내부의 모든 DRAM 칩이 1개의 chip select 신호를 공유하는 구조이며, 각 DRAM 칩은 동일한 동작을 수행한다. 쓰레디드 메모리 모듈은 chip select 신호를 2개, 4개, 또는 8개를 사용하여 독립적인 동작이 가능한 DRAM 칩의 개수(메모리 쓰레드 개수)를 증가시킨다. [그림 1]은 메모리 쓰레드 개수가 8개인 쓰레디드 메모리 모듈을 나타낸 것이다. 독립적인 DRAM 칩의 개수가 증가하면, 사용가능한 DRAM 페이지 버퍼 개수가 증가하여 버퍼 hit 확률을 높이는 효과가 있다. 메모리 접근 요청의 지역성(locality)이 낮은 경우, PRECHARGE-ACTIVATE 횟수를 줄임으로써 접근 시간과 전력 소모를 줄일 수 있다. 반면, 각 메모리 쓰레드가 사용하는 데이터 버스

폭이 줄어들기 때문에 전송시간이 증가하는 단점이 있다.

## 2.2 메모리 접근 시간

메모리 접근으로 인한 지연 시간을 대기 시간과 DRAM 접근 시간, 전송 시간으로 나누어 살펴 볼 수 있다. 대기시간( $T_Q$ )은 캐쉬 등에 의해 메모리 접근 요청이 이루어진 시점부터 메모리 모듈에 DRAM 명령을 보내기 전까지 메모리 컨트롤러의 버퍼에서 기다리는 시간이다. 각 프로세서에서 메인 메모리에 접근하는 주기가 일정하다고 가정하면, 프로세서 코어의 개수가 증가할수록 대기 시간은 증가한다. 메모리 쓰레드 개수가 증가하면, DRAM 명령의 파이프라이닝을 통해 동시에 수행할 수 있는 메모리 접근 요청의 개수가 증가하므로 대기시간을 줄일 수 있다. DRAM 접근 시간( $T_L$ )은 tRC, tCAS 등의 DRAM 타이밍 파라미터와 DRAM 페이지 버퍼의 상태에 의해 결정된다. 메모리 쓰레드 개수가 증가 할수록, 사용할 수 있는 DRAM 페이지 버퍼의 hit 확률이 증가하여 DRAM 접근 시간은 감소한다. 전송시간( $T_X$ )은 각 메모리 접근 요청에 의해 데이터 버스가 사용되는 시간이다. 메모리 쓰레드 개수가 증가할수록, 전송시간은 비례하여 증가한다.

## III. 실험 결과

프로세서 개수와 메모리 쓰레드 개수에 따른 메모리 접근 성능을 시뮬레이션을 통해 비교하였다. 시뮬레이션은 컴퓨터 아키텍처 시뮬레이터인 M5 [3]를 사용하여 execution-driven 시뮬레이션을 수행하였다. 메인 메모리 시뮬레이션 모델은 DDR3를 기준으로 메모리 접근 타이밍을 계산하고, 어드레스 매핑과 메모리 접근 스케줄링을 수행한다. 메모리 주소를 상위 비트부터 (row, bank, thread, col)순으로 매핑하였으며, FR-FCFS (First Ready-First Come First Serve) 방식의 메모리 접근 스케줄링 방식을 사용하였다.

각 프로세서가 512x512 행렬과 벡터의 곱셈 연산을 수행하는 동안의 평균 메모리 접근 시간은 [그림 2]와 같다. 메모리 쓰레드가 1개인 기존의 메모리 모듈(1x64)은 프로세서 개수가 증가함에 따라 대기 시간이 증가하여 메모리 접근 지연 시간이 증가한다. 메모리 쓰레드를 2개로 증가시킬 경우, 대기시간이 감소되지만 전송 시간의 증가로 그 효과가 상쇄된다. 메모리 쓰레드 개수가 4개와 8개일 경우, 대기시간이 급격하게 증가하여 메모리 접근 지연 시간 또한 크게 증가한다. DDR3와 같이 BL8 모드만 지원하는 경우, 쓰레드 메모리 모듈은 한 개의 캐쉬 라인 요청에 대하여

복수개의 READ/WRITE 명령이 필요하다. 특히, 메모리 쓰레드 개수가 8개일 경우, addr/cmd 버스가 포화상태가 되어 메모리 접근 지연 시간이 크게 증가한다.

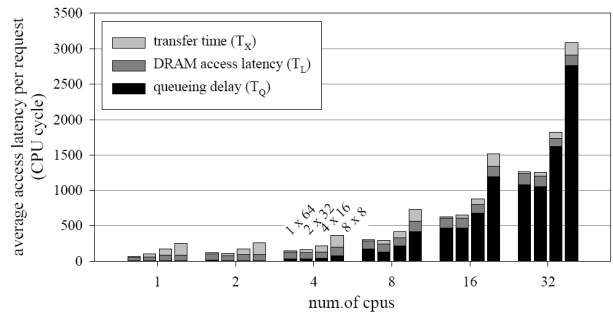


그림 2. 평균 메모리 접근 지연 시간

## IV. 결론 및 향후 연구 방향

쓰레드 메모리 모듈은 독립적인 동작이 가능한 DRAM의 개수를 증가시켜서 복수개의 메모리 접근 요청을 처리할 수 있도록 한다. 본 연구에서 사용한 FR-FCFS 스케줄링 방식은 DRAM 접근 시간( $T_L$ )을 줄이는 방향으로 메모리 접근 스케줄링을 수행한다. 쓰레드 메모리 모듈을 사용하여 접근 지연 시간을 줄이기 위해서는 대기시간( $T_Q$ )을 줄일 수 있는 메모리 접근 스케줄링 방식과 메모리 접근 요청을 각 메모리 쓰레드에 고르게 분산시킬 수 있는 어드레스 매핑 방식이 필요하다.

이 논문은 지식경제부 출연금으로 ETRI와 시스템반도체산업진흥센터에서 수행한 ITSoc 핵심설계인력양성사업과 교육과학기술부의 재원으로 한국학술진흥재단에서 수행하는 BK21 프로젝트의 지원을 받아 수행된 연구입니다.

## 참고문헌

- [1] F. A. Ware and C. Hampel, "Improving Power and Data Efficiency with Threaded Memory Modules," International Conference on Computer Design (ICCD), 2006.
- [2] J. H. Ahn, H. Levenrich, R. S. Schreiber, and N. P. Jouppi, "Multicore DIMM: an Energy Efficient Memory Module with Independently Controlled DRAMs," IEEE Computer Architecture Letters, 2008.
- [3] N. L. Binkert, R. G. Dreslinski, L. R. Hsu, K. T. Lim, A. G. Saidi and S. K. Reinhardt, "The M5 Simulator: Modeling Networked Systems," IEEE Micro, 2006.